REJ10B0116-0200

Everywhere you imagine.

SuperH™ Family E10A-USB Emulator

Additional Document for User's Manual

Supplementary Information on Using the SH7750R

Renesas Microcomputer Development Environment System

SuperH™ Family / SH7750 Series

E10A-USB for SH7750R HS7750RKCU01HE

User's Manual

Rev.2.00
Revision Date: Mar. 19, 2007

Renesas Technology
www.renesas.com

Notes regarding these materials

1. This document is provided for reference purposes only so that Renesas customers may select the appropriate Renesas products for their use. Renesas neither makes warranties or representations with respect to the accuracy or completeness of the information contained in this document nor grants any license to any intellectual property rights or any other rights of Renesas or any third party with respect to the information in this document.
2. Renesas shall have no liability for damages or infringement of any intellectual property or other rights arising out of the use of any information in this document, including, but not limited to, product data, diagrams, charts, programs, algorithms, and application circuit examples.
3. You should not use the products or the technology described in this document for the purpose of military applications such as the development of weapons of mass destruction or for the purpose of any other military use. When exporting the products or technology described herein, you should follow the applicable export control laws and regulations, and procedures required by such laws and regulations.
4. All information included in this document such as product data, diagrams, charts, programs, algorithms, and application circuit examples, is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas products listed in this document, please confirm the latest product information with a Renesas sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas such as that disclosed through our website. (http://www.renesas.com)
5. Renesas has used reasonable care in compiling the information included in this document, but Renesas assumes no liability whatsoever for any damages incurred as a result of errors or omissions in the information included in this document.
6. When using or otherwise relying on the information in this document, you should evaluate the information in light of the total system before deciding about the applicability of such information to the intended application. Renesas makes no representations, warranties or guaranties regarding the suitability of its products for any particular application and specifically disclaims any liability arising out of the application and use of the information in this document or Renesas products.
7. With the exception of products specified by Renesas as suitable for automobile applications, Renesas products are not designed, manufactured or tested for applications or otherwise in systems the failure or malfunction of which may cause a direct threat to human life or create a risk of human injury or which require especially high quality and reliability such as safety systems, or equipment or systems for transportation and traffic, healthcare, combustion control, aerospace and aeronautics, nuclear power, or undersea communication transmission. If you are considering the use of our products for such purposes, please contact a Renesas sales office beforehand. Renesas shall have no liability for damages arising out of the uses set forth above.
8. Notwithstanding the preceding paragraph, you should not use Renesas products for the purposes listed below:
   (1) artificial life support devices or systems
   (2) surgical implantations
   (3) healthcare intervention (e.g., excision, administration of medication, etc.)
   (4) any other purposes that pose a direct threat to human life

   Renesas shall have no liability for damages arising out of the uses set forth in the above and purchasers who elect to use Renesas products in any of the foregoing applications shall indemnify and hold harmless Renesas Technology Corp., its affiliated companies and their officers, directors, and employees against any and all damages arising out of such applications.
9. You should use the products described herein within the range specified by Renesas, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas shall have no liability for malfunctions or damages arising out of the use of Renesas products beyond such specified ranges.
10. Although Renesas endeavors to improve the quality and reliability of its products, IC products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Please be sure to implement safety measures to guard against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other applicable measures. Among others, since the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
11. In case Renesas products listed in this document are detached from the products to which the Renesas products are attached or affixed, the risk of accident such as swallowing by infants and small children is very high. You should implement safety measures so that Renesas products may not be easily detached from your products. Renesas shall have no liability for damages arising out of such detachment.
12. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written approval from Renesas.
13. Please contact a Renesas sales office if you have any questions regarding the information contained in this document, Renesas semiconductor products, or if you have any other inquiries.

Contents

Section 1 Connecting the Emulator with the User System 1
1.1 Components of the Emulator 1
1.2 Connecting the E10A-USB Emulator with the User System 2
1.3 Installing the H-UDI Port Connector on the User System 2
1.4 Pin Assignments of the H-UDI Port Connector 2
1.5 Recommended Circuit between the H-UDI Port Connector and the MPU 4
1.5.1 Recommended Circuit (14-Pin Type) 4

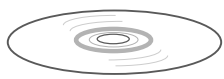
Section 2 Specifications of the Software when Using the SH7750R 7
2.1 Differences between the SH7750R and the Emulator 7
2.2 Specific Functions for the Emulator when Using the SH7750R 11
2.2.1 Break Condition Functions 11
2.2.2 Trace Functions 13
2.2.3 Notes on Using the JTAG Clock (TCK) 16
2.2.4 Notes on Setting the [Breakpoint] Dialog Box 16
2.2.5 Notes on Setting the [Break Condition] Dialog Box and the BREAKCONDITION_ SET Command 18
2.2.6 Performance Measurement Functions 20
2.2.7 Interrupts 27
2.2.8 CPU Status Acquisition 28

Section 1 Connecting the Emulator with the User System

1.1 Components of the Emulator

The E10A-USB emulator supports the SH7750R. Table 1.1 lists the components of the emulator.

Table 1.1 Components of the Emulator

Classification	Component	Appearance	Quantity	Remarks
Hardware	Emulator box		1	HS0005KCU01H: Depth: 65.0 mm, Width: 97.0 mm, Height: 20.0 mm, Mass: 72.9 g or HS0005KCU02H: Depth: 65.0 mm, Width: 97.0 mm, Height: 20.0 mm, Mass: 73.7 g
	User system interface cable		1	14-pin type: Length: 20 cm, Mass: 33.1 g
	USB cable		1	Length: 150 cm, Mass: 50.6 g
Software	SH7750R E10A-USB emulator setup program, SuperH™ Family E10A-USB Emulator User's Manual, Supplementary Information on Using the SH7750R*, and Test program manual for HS0005KCU01H and HS0005KCU02H		1	HS0005KCU01SR, HS0005KCU01HJ, HS0005KCU01HE, HS7750RKCU01HJ, HS7750RKCU01HE, HS0005TM01HJ, and HS0005TM01HE (provided on a CD-R)

Note: Additional document for the MPUs supported by the emulator is included. Check the target MPU and refer to its additional document.

1.2 Connecting the E10A-USB Emulator with the User System

To connect the E10A-USB emulator (hereinafter referred to as the emulator), the H-UDI port connector must be installed on the user system to connect the user system interface cable. When designing the user system, refer to the recommended circuit between the H-UDI port connector and the MCU. In addition, read the E10A-USB emulator user's manual and hardware manual for the related device.

1.3 Installing the H-UDI Port Connector on the User System

Table 1.2 shows the recommended H-UDI port connectors for the emulator.

Table 1.2 Recommended H-UDI Port Connectors

Connector	Type Number	Manufacturer	Specifications
14-pin connector	2514-6002	Minnesota Mining & Manufacturing Ltd.	14-pin straight type

Note: Do not place any components within 3 mm of the H-UDI port connector.

1.4 Pin Assignments of the H-UDI Port Connector

Figure 1.1 shows the pin assignments of the H-UDI port connector.

Note: Note that the pin number assignments of the H-UDI port connector shown on the following page differ from those of the connector manufacturer.

Pin No.	Signal		Input/ Output*1	SH7750R Pin No. BGA256	SH7750R Pin No. HQFP208	Note
1	TCK		Input	A-5	198	
2	/TRST	*2	Input	C-4	200	
3	TDO		Output	A-6	194	
4	/ASEBRK BRKACK	*2	Output	B-7	193	
5	TMS		Input	B-6	197	
6	TDI		Input	B-5	199	
7	/RESETP	*2	Output	B-1	2	User reset
8	N.C.	*5	—	—	—	
9	GND		—	—	—	
11	UVCC	*4	Output	—	—	
10, 12, and 13	GND		—	—	—	
14	GND	*3	Output	—	—	

Notes: 1. Input to or output from the user system.
2. The slash (/) means that the signal is active-low.
3. The emulator monitors the GND signal of the user system and detects whether or not the user system is connected.
4. If the VccQ pin is not connected to the UVCC, the I/O voltage of the user system interface will be fixed to 3.3 V.
5. This pin can be connected to GND.

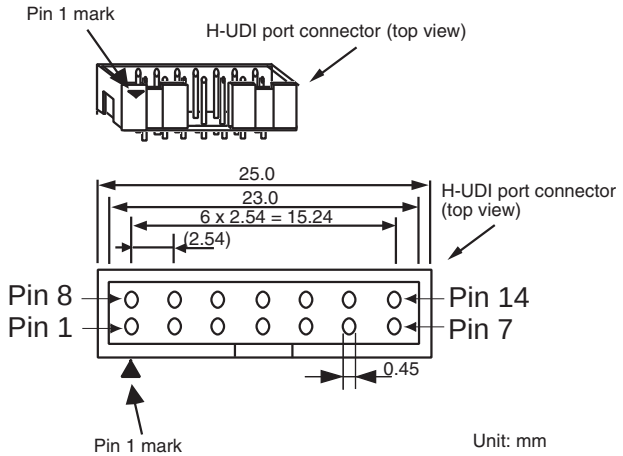


Figure 1.1 Pin Assignments of the H-UDI Port Connector

1.5 Recommended Circuit between the H-UDI Port Connector and the MPU

1.5.1 Recommended Circuit (14-Pin Type)

Figure 1.2 shows a recommended circuit for connection between the H-UDI port connector and the MPU when the emulator is in use. Figure 1.3 shows a circuit for connection when UVCC is not connected.

Notes: 1. Do not connect anything to the N.C. pins of the H-UDI port connector.

2. When a network resistance is used for pull-up, it may be affected by a noise. Separate TCK from other resistances.
3. The pattern between the H-UDI port connector and the MPU must be as short as possible. Do not connect the signal lines to other components on the board.
4. When the power supply of the user system is turned off, supplying VccQ of the user system to the UVCC pin reduces the leakage current from the emulator to the user system. A level shifter that is activated by the internal power supply or user power supply (changed by the switch) is installed in the interface circuit of the emulator. If the user power is supplied to the UVCC pin, the level shifter is not activated as long as no user power is supplied. When the power supply of the user system is turned off, no current flows from the user interface. The I/O voltage level of the user system interface can be the same as that of the VccQ. To operate the emulator with low voltage (lower than 3.3 V), the VccQ must be supplied to the UVCC pin. Make the emulator's switch settings so that the VccQ will be supplied (SW2 = 1 and SW3 = 1) (as shown in figure 1.2).
5. The resistance values shown in figures 1.2 and 1.3 are recommended.
6. For the pin processing in cases where the emulator is not used, refer to the hardware manual of the related MPU.

When the circuit is connected as shown in figure 1.2, the switches of the emulator are set as SW2 = 1 and SW3 = 1. For details, refer to section 3.8, Setting the DIP Switches, in the SuperH™ Family E10A-USB Emulator User's Manual.

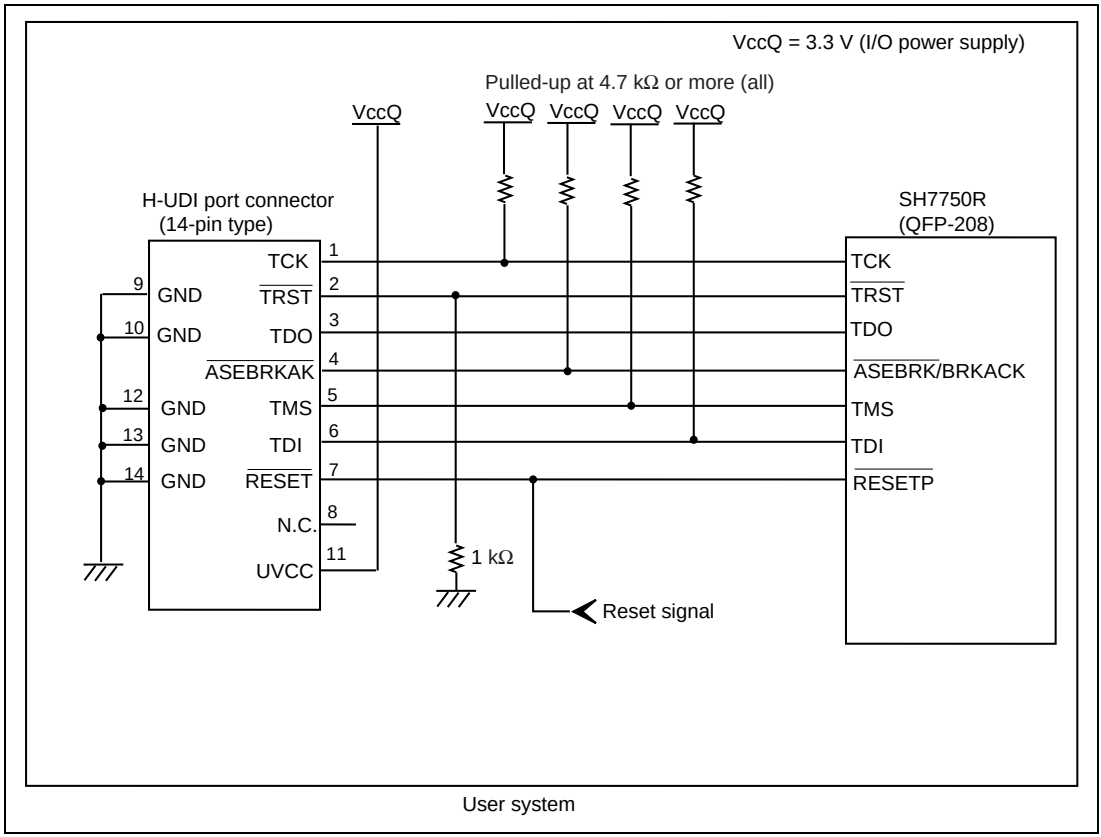


Figure 1.2 Recommended Circuit for Connection between the H-UDI Port Connector and MPU when the Emulator is in Use (14-Pin Type UVCC Connected)

When the circuit is connected as shown in figure 1.3, the switches of the emulator are set as SW2 = 0 and SW3 = 1. For details, refer to section 3.8, Setting the DIP Switches, in the SuperH™ Family E10A-USB Emulator User's Manual.

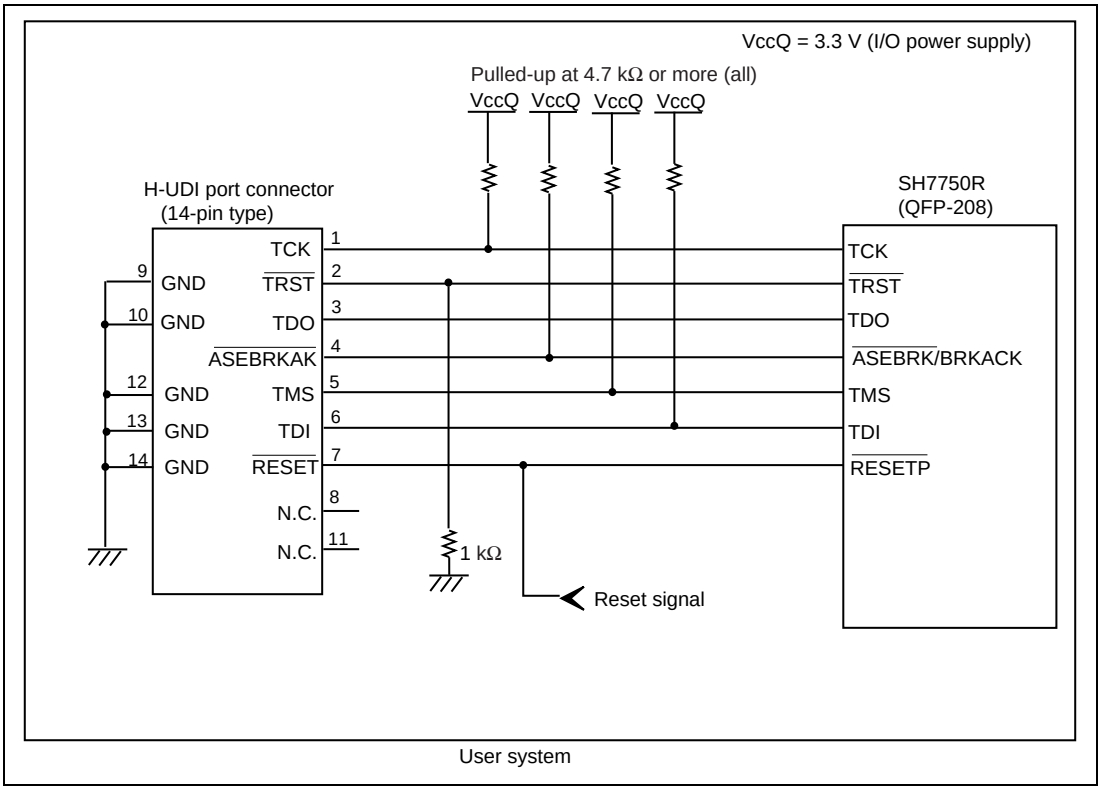


Figure 1.3 Circuit for Connection between the H-UDI Port Connector and MPU when the Emulator is in Use (14-Pin Type UVCC Not Connected*)

Note: When UVCC is not connected and the user system is turned off, note that the leakage current flows from the emulator to the user system.

Section 2 Specifications of the Software when Using the SH7750R

2.1 Differences between the SH7750R and the Emulator

1. When the emulator system is initiated, it initializes the general registers and part of the control registers as shown in table 2.1. When the emulator is initiated from the workspace, a value to be entered is saved in a session.

Table 2.1 Register Initial Values at Emulator Link Up

Register	Emulator at Link Up
R0 to R14	H'00000000
R15 (SP)	H'00000000
R0_BANK to R7_BANK	H'00000000
PC	H'A0000000
SR	H'700000F0
GBR	H'00000000
VBR	H'00000000
MACH	H'00000000
MACL	H'00000000
PR	H'00000000
DBR	H'00000000
SGR	H'00000000
SPC	H'00000000
SSR	H'000000F0
FPUL	H'00000000
FPSCR	H'00040001
FR0 to FR15	H'00000000
XF0 to XF15	H'00000000

2. The emulator uses the H-UDI; do not access the H-UDI.

3. Low-Power States (Sleep, Standby, and Module Standby)

   For low-power consumption, the SH7750R has sleep, standby, and module standby modes.

   The sleep and standby modes are switched using the SLEEP instruction. When the emulator is used, the sleep and standby modes can be cleared by either normal clearing or by the [STOP] button. In the latter case, the user program breaks. Note, however, that if a command has been entered in standby mode or module standby mode, a TIMEOUT error will occur.

Notes: 1. After the sleep mode is cleared by a break, execution restarts at the instruction following the SLEEP instruction.

2. If the memory is accessed or modified in sleep mode, the sleep mode is cleared and execution starts at the instruction following the SLEEP instruction.

3. If an operation such as the command input is performed in the hardware standby state, the command from the emulator cannot be used, or the state cannot be cancelled by the [STOP] button.

4. When the SLEEP instruction is executed by a step command and [Step…] in the [Run] menu is used, set [Rate] as 6. If 5 or lower value is set, a communication timeout error will occur.

4. Reset Signals

   The SH7750R reset signal is only valid during emulation started with clicking the GO or STEP-type button. If this signal is enabled on the user system in command input wait state, it is not sent to the SH7750R.

Note: Do not break the user program when the /RESET, /MRESET, or /BREQ signal is being low or the /RDY signal is being high. A TIMEOUT error will occur. If the /BREQ signal is fixed to low or the /RDY signal is fixed to high during break, a TIMEOUT error will occur at memory access.

5. Direct Memory Access Controller (DMAC)

   The DMAC operates even when the emulator is used. When a data transfer request is generated, the DMAC executes DMA transfer.

6. Memory Access during User Program Execution

   When a memory is accessed from the memory window, etc. during user program execution, the user program is resumed after it has stopped in the emulator to access the memory. Therefore, realtime emulation cannot be performed.

The stopping time of the user program is as follows:

Environment:

Host computer: 1 GHz (Pentium® III)
OS: Windows® 2000
SH7750R: 267 MHz (CPU clock)
JTAG clock: 20 MHz

When a one-byte memory is read from the command-line window, the stopping time will be about 8 ms.

7. Interrupt

When the NMIB bit in the ICR register is 1, the NMI interrupt is accepted during break and the program is executed from the NMI interrupt vector. If the program cannot return normally from the NMI interrupt routine or the value in the general-purpose register is not guaranteed, a communication timeout error will occur.

8. Memory Access during User Program Break

The emulator can download the program for the flash memory area (refer to section 6.22, Download Function to the Flash Memory Area, in the SuperH™ Family E10A-USB Emulator User's Manual). Other memory write operations are enabled for the RAM area. Therefore, an operation such as memory write or BREAKPOINT should be set only for the RAM area. When the memory area can be written by the MMU, do not perform memory write, BREAKPOINT break, or downloading.

9. Cache Operation during User Program Break

When cache is enabled, the emulator accesses the memory by the following methods:

- At memory write: Writes through the cache, then writes to the memory.
- At memory read: Does not change the cache write mode that has been set.

Therefore, when memory read or write is performed during user program break, the cache state will be changed.
When the half of operand cache is used as an internal RAM and memory fill is performed in this area, the verify option must be disabled. Memory fill is not performed correctly if the verify option is enabled.

10. UBC

When [User] is specified in the [UBC mode] list box in the [Configuration] dialog box, the UBC can be used in the user program.
Do not use the UBC in the user program as it is used by the emulator when [EML] is specified in the [UBC mode] list box in the [Configuration] dialog box.

11. Loading Sessions

    Information in [JTAG clock] of the [Configuration] dialog box cannot be recovered by loading sessions. Thus the TCK value will be 0.625 MHz.

12. [IO] Window

- Display and modification

  Do not change values of the User Break Controller because it is used by the emulator.

  For each watchdog timer register, there are two registers to be separately used for write and read operations.

Table 2.2 Watchdog Timer Register

Register Name	Usage	Register
WTCSR(W)	Write	Watchdog timer control/status register
WTCNT(W)	Write	Watchdog timer counter
WTCSR(R)	Read	Watchdog timer control/status register
WTCNT(R)	Read	Watchdog timer counter

- The watchdog timer operates only when the user program is executed. Do not change the value of the frequency change register in the [IO] window or [Memory] window.
- The internal I/O registers can be accessed from the [IO] window. However, note the following when accessing the SDMR register of the bus-state controller. Before accessing the SDMR register, specify addresses to be accessed in the I/O-register definition file (SH7750R.IO) and then activate the High-performance Embedded Workshop. After the I/O-register definition file is created, the MPU's specification may be changed. If each I/O register in the I/O-register definition file differs from addresses described in the hardware manual, change the I/O-register definition file according to the description in the hardware manual. The I/O-register definition file can be customized depending on its format. Note that, however, the emulator does not support the bit-field function.
- Verify

  In the [IO] window, the verify function of the input value is disabled.

13. Illegal Instructions

    If illegal instructions are executed by STEP-type commands, the emulator cannot go to the next program counter.

2.2 Specific Functions for the Emulator when Using the SH7750R

2.2.1 Break Condition Functions

In addition to BREAKPOINT functions, the emulator has Break Condition functions. Eight types of conditions can be set under Break Condition 1, 2, 3, 4, 5, 6, 7, 8. Break Condition 5,6 use the user break controller (UBC). Table 2.3 lists these conditions of Break Condition.

Table 2.3 Types of Break Conditions

Break Condition Type	Description
Address bus condition (Address)	Breaks when the SH7750R address bus value or the program counter value matches the specified value.
Data bus condition (Data)	Breaks when the SH7750R data bus value matches the specified value. Byte, word, or longword can be specified as the access data size.
ASID condition (ASID)	Breaks when the SH7750R ASID value matches the specified condition.
Bus state condition (Bus State)	There are two bus state condition settings: Read/write condition: Breaks in the read or write cycle of the SH7750R. Bus state condition: Breaks when the operating state in an SH7750R bus cycle matches the specified condition.
LDTLB instruction break condition	Breaks when the SH7750R executes the LDTLB instruction.
Internal I/O break condition	Breaks when the SH7750R accesses the internal I/O.

Note: For the window function or command-line syntax, refer to the online help.

Table 2.4 lists the combinations of conditions that can be set under Break Condition 1, 2, 3, 4, 5, 6, 7, 8.

Table 2.4 Dialog Boxes for Setting Break Conditions

	Dialog Box			
	[Break Condition 1, 5] Dialog Box	**[Break Condition 2, 3, 4, 6] Dialog Box**	**[Break Condition 7] Dialog Box**	**[Break Condition 8] Dialog Box**
Address bus condition (Address)	O	O	X	X
Data bus condition (Data)	O	X	X	X
ASID condition (ASID)	O	O	X	X
Read/write specification	O	O	X	X
Data access	O	O	X	X
Before/after execution	O	O	X	X
Sequential break	O	O	X	X
LDTLB instruction break	X	X	X	O
Internal I/O access break	X	X	O	X

Note: O: Can be set in the dialog box.
X: Cannot be set in the dialog box.

Note: If the BL bit of the SR register is 1, do not use BREAKPOINTs.
If a break is specified for an address that is close to an address whose instruction generates a manual reset, a manual reset may be generated instead of a break. Therefore, to ensure the performance of a break, specify a break for an address that is four addresses before the address whose instruction generates an exception.

The emulator has sequential break functions. Table 2.5 shows the sequential break conditions.

Table 2.5 Sequential Break Conditions

No.	Break Condition	Description
1	Sequential break condition 2-1	Program is halted when Break Condition 2 and Break Condition 1 are satisfied in that order. Break Condition 2,1 should be set.
2	Sequential break condition 3-2-1	Program is halted when Break Condition 3, Break Condition 2, and Break Condition 1 are satisfied in that order. Break Condition 3,2,1 should be set.
3	Sequential break condition 4-3-2-1	Program is halted when Break Condition 4, Break Condition 3, Break Condition 2, and Break Condition 1 are satisfied in that order. Break Condition 4,3,2,1 should be set.
4	Sequential break condition 6-5	Program is halted when Break Condition 6 and Break Condition 5 are satisfied in that order. Break Condition 6,5 should be set.

2.2.2 Trace Functions

The emulator does not support the AUD function.

Table 2.6 shows the internal trace functions.

Table 2.6 Internal Trace Functions

Function	Description
Branch instruction trace	Traces and displays the branch instructions. The branch source address and branch destination address for the eight latest branch instructions are displayed. There are three kinds of branch instruction trace: • Normal branch instruction trace Traces and displays the normal branch instructions. The normal branch instructions are the BF, BF/S, BT/S, BRA, BRAF, and JMP instructions. To use this function, select the [Acquire normal branch instruction trace] check box of the [Branch trace] page. • Subroutine branch instruction trace Traces and displays the subroutine branch instructions. The subroutine branch instructions are the BSR, BSRF, JSR, and RTS instructions. To use this function, select the [Acquire subroutine branch instruction trace] check box of the [Branch trace] page • Exception branch instruction trace Traces and displays the exception branch instruction. The exception branch instruction is the RTE instruction. In addition, all the exception and interrupt operations are traced. To use this function, select the [Acquire exceptional branch instruction trace] check box of the [Branch trace] page.
Continous trace	Acquires the trace information continuously. This is called continuous trace. For the branch instruction trace, eight-branch information can be repeatedly acquired a maximum of four times. Select the [Acquire continuous trace] check box of the [Branch trace] page. If continuous trace is selected, realtime trace cannot be performed.
Internal I/O trace	Traces and displays the address and data that access the internal I/O area. To use this function, select the [Get trace information of internal I/O Area] radio button in the [Break Condition 7] dialog box and the [Acquire continuous trace] check box of the [Branch trace] page.
LDTLB instruction execution trace	Traces and displays the address that executes the LDTLB instruction. To use this function, select the [Get trace information of LDTLB instruction] radio button in the [Break Condition 8] dialog box and the [Acquire continuous trace] check box of the [Branch trace] page.

Notes: 1. When the continuous trace is not used, trace acquisition of the eight latest branch instructions is enabled.

2. If an interrupt is generated at the program execution start or end, including a step operation, the emulator address may be acquired. In such a case, the following message will be displayed. Ignore this address because it is not a user program address.

   *** EML ***

3. If a completion-type exception occurs during exception branch acquisition, the next address to the address in which an exception occurs is acquired.
4. When a user interrupt is enabled by the INTERRUPT command during the emulator command wait state or user program execution, an interrupt that is generated at the program execution start or end, including a step operation, can be traced in realtime.
5. When the [Acquire continuous trace] check box is selected, do not perform memory access during emulation.
6. When internal I/O trace or LDTLB instruction trace is performed, select the [Acquire continuous trace] check box.
7. When the [Acquire continuous trace] check box is selected, 32 trace information data can be acquired. In this case, however, since the user program stops at constant intervals, the processing speed is decreased compared with the case where the [Acquire continuous trace] check box is not selected
8. Trace information cannot be acquired for the following branch instructions:
   — The BF and BT instructions whose displacement value is 0
   — Branch to H'A0000000 by reset
9. When the [Acquire continuous trace] check box is selected, and when either the [Get trace information of internal I/O area] radio button (internal I/O trace enabled) or the [Get trace information of LDTLB instruction] radio button is selected (LDTLB instruction trace enabled) with the [Break Condition 5] dialog box,
   — An internal I/O trace cannot be made with the Step In command.
   — The LDTLB instruction and internal I/O trace cannot be performed with the Step Over command.
10. When continuous trace is used, do not enable user interrupt by the INTERRUPT command during the emulator command wait state or user program execution.
11. Do not use the continuous trace for a program in which an SGR value is referred to with the interrupt handler. In the emulator, the contents of the SGR register are lost when the user program breaks. Since the user program execution stops at constant intervals while the continuous trace is selected, the contents of the SGR register will be lost.

2.2.3 Notes on Using the JTAG Clock (TCK)

Set the JTAG clock (TCK) frequency to lower than the frequency of the SH7750R peripheral module clock (CKP).

2.2.4 Notes on Setting the [Breakpoint] Dialog Box

1. When an odd address is set, the next lowest even address is used.
2. A BREAKPOINT is accomplished by replacing instructions. Accordingly, it can be set only to the internal RAM area. However, a BREAKPOINT cannot be set to the following addresses:
   - An address whose memory content is H'003B
   - An area other than the CS0 to CS6 areas and the internal RAM area
   - An instruction in which Break Condition 3 is satisfied
   - A slot instruction of a delayed branch instruction

   In addition, do not perform memory write, BREAKPOINT, or download even if the memory space can only be written by the MMU.
3. During step operation, a BREAKPOINT is disabled.
4. Conditions set at Break Condition 3 are disabled when an instruction to which a BREAKPOINT has been set is executed. Do not set a BREAKPOINT to an instruction in which Break Condition 3 is satisfied.
5. When execution resumes from the address where a BREAKPOINT is specified, single-step operation is performed at the address before execution resumes. Therefore, realtime operation cannot be performed.
6. When a BREAKPOINT is set to the slot instruction of a delayed branch instruction, the PC value becomes an illegal value. Accordingly, do not set a BREAKPOINT to the slot instruction of a delayed branch instruction.
7. When the [Normal] option is selected in the [Memory area] group box in the [General] page of the [Configuration] dialog box, a BREAKPOINT is set to a physical address or a virtual address according to the SH7750R MMU state during command input when the VPMAP_SET command setting is disabled. The ASID value of the SH7750R PTEH register during command input is used. When VPMAP_SET command setting is enabled, a BREAKPOINT is set to a physical address into which address translation is made according to the VP_MAP table. However, for addresses out of the range of the VP_MAP table, the address to which a BREAKPOINT is set depends on the SH7750R MMU state during command input. Even when the VP_MAP table is modified after BREAKPOINT setting, the address translated at BREAKPOINT setting is valid.
8. When the [Physical] option is selected in the [Memory area] group box in the [General] page of the [Configuration] dialog box, a BREAKPOINT is set to a physical address. A

BREAKPOINT is set after disabling the SH7750R MMU during program execution. After setting, the MMU is returned to the original state. When a break occurs at the corresponding virtual address, the cause of termination displayed in the status bar and the [Status] window is ILLEGAL INSTRUCTION, not BREAKPOINT.

9. When the [Virtual] option is selected in the [Memory area] group box in the [General] page of the [Configuration] dialog box, a BREAKPOINT is set to a virtual address. A BREAKPOINT is set after enabling the SH7750R MMU during program execution. After setting, the MMU is returned to the original state. When an ASID value is specified, the BREAKPOINT is set to the virtual address corresponding to the ASID value. The emulator sets the BREAKPOINT after rewriting the ASID value to the specified value, and returns the ASID value to its original value after setting. When no ASID value is specified, the BREAKPOINT is set to a virtual address corresponding to the ASID value at command input.
10. If a TLB error occurs during virtual address setting, the following message box will be displayed.

Figure 2.1 Message Box for Clearing a TLB-Error

If a program is executed again without clearing the BREAKPOINT set at the address in which the TLB error occurs, a TLB error will occur again. Accordingly, clear the BREAKPOINT before execution.

11. An address (physical address) to which a BREAKPOINT is set is determined when the BREAKPOINT is set. Accordingly, even if the VP_MAP table is modified after BREAKPOINT setting, the BREAKPOINT address remains unchanged. When a BREAKPOINT is satisfied with the modified address in the VP_MAP table, the cause of termination displayed in the status bar and the [Status] window is ILLEGAL INSTRUCTION, not BREAKPOINT.
12. When a BREAKPOINT is set to the cacheable area, the cache block containing the BREAKPOINT address is filled immediately before and after user program execution.
13. While a BREAKPOINT is set, the contents of the instruction cache are disabled at execution completion.

14. If an address of a BREAKPOINT cannot be correctly set in the ROM or flash memory area, a mark ● will be displayed in the [BP] area of the address on the [Source] or [Disassembly] window by refreshing the [Memory] window, etc. after Go execution. However, no break will occur at this address. When the program halts with the break condition, the mark ● disappears.

2.2.5 Notes on Setting the [Break Condition] Dialog Box and the BREAKCONDITION_SET Command

1. When [Go to cursor], [Step In], [Step Over], or [Step Out] is selected, the settings of Break Condition 3 are disabled.
2. Break Condition 3 is disabled when an instruction to which a BREAKPOINT has been set is executed. Accordingly, do not set a BREAKPOINT to an instruction which satisfies Break Condition 3.
3. When a Break Condition is satisfied, emulation may stop after two or more instructions have been executed.
4. If a PC break address condition is set to the slot instruction after a delayed branch instruction, user program execution cannot be terminated before the slot instruction execution; execution stops before the branch destination instruction.
5. Break Condition 5,6 use the UBC. When the UBC is used in the user program, change the UBC setting for users by using the [UBC_mode] list box in the [Configuration] dialog box or the UBC_mode command.
6. Break Condition 1,4 are used as the start and end conditions of the performance measurement function when [PA-1 start point] and [PA-1 end point] are displayed on the [Action] part in the [Break condition] sheet of the [Event] window. For details on setting the performance measurement function, refer to section 2.2.6, Performance Measurement Functions.
   This applies when the Break Condition is displayed with the BREAKCONDITION_DISPLAY command in the command-line function. In this case, a break does not occur when Break Condition 1,4 are satisfied.

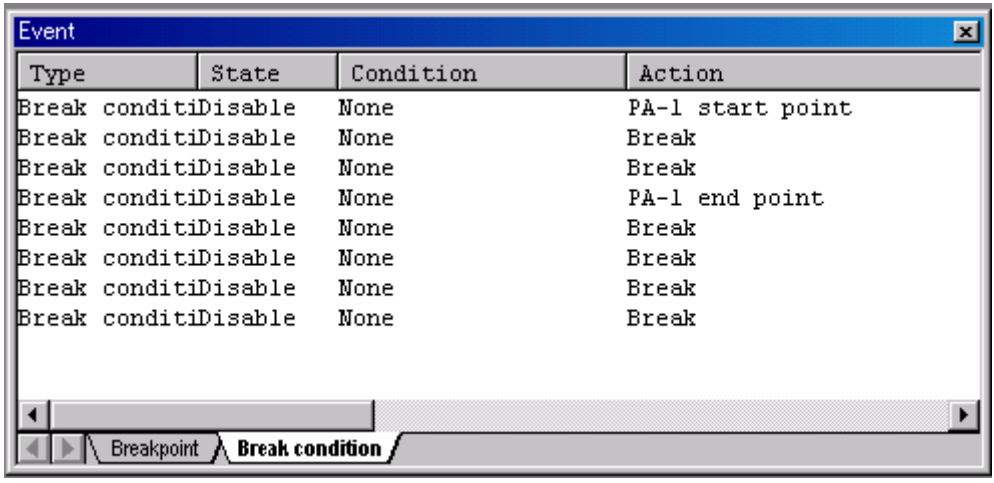


Figure 2.2 [Event] Window

2.2.6 Performance Measurement Functions

The emulator supports the performance measurement function.

1. Setting the performance measurement conditions

   To set the performance measurement conditions, use the [CPU Performance] dialog box and the PERFORMANCE_SET command. When any line on the [Performance Analysis] window is clicked with the right mouse button, the popup menu is displayed and the [CPU Performance] dialog box is displayed by selecting [Setting].

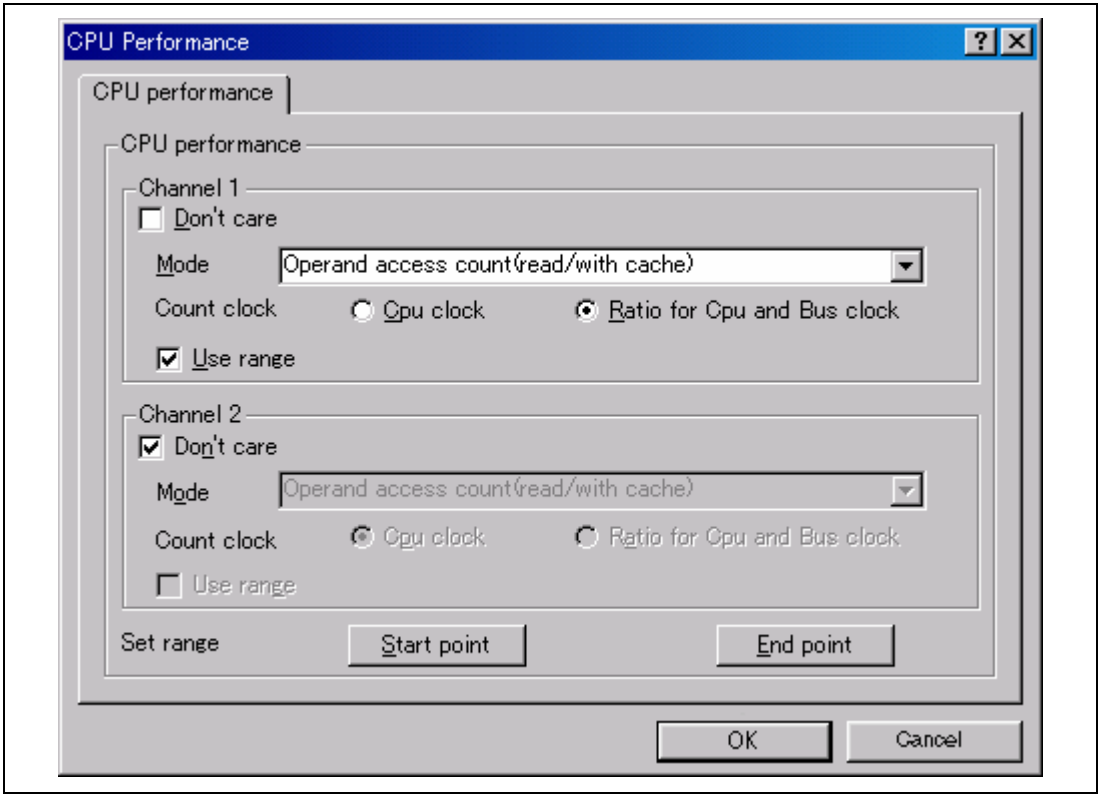


Figure 2.3 [CPU Performance] Dialog Box

Note: For the command line syntax, refer to the online help.

The emulator measures how many times the conditions of the user program specified with the performance analysis function are satisfied. For this function, two events can be measured simultaneously and the following conditions can be specified.

(a) Measurement range

One of the following ranges can be specified by either of measurement channels 1 and 2.

1. From the start to the end of the user program execution
2. From the satisfaction of the condition set in Break Condition 1 to the satisfaction of the condition set in Break Condition 4

When the first range is specified, the measurement result includes a several-cycle error for one user program execution. Therefore, do not specify this range when the step is to be executed. In addition, the user program execution stops when continuous trace is used; again, do not specify the first range in this case.

In the second range, [PA-1 start point] and [PA-1 end point] are displayed on the [Action] part in the [Break condition] sheet of the [Event] window.

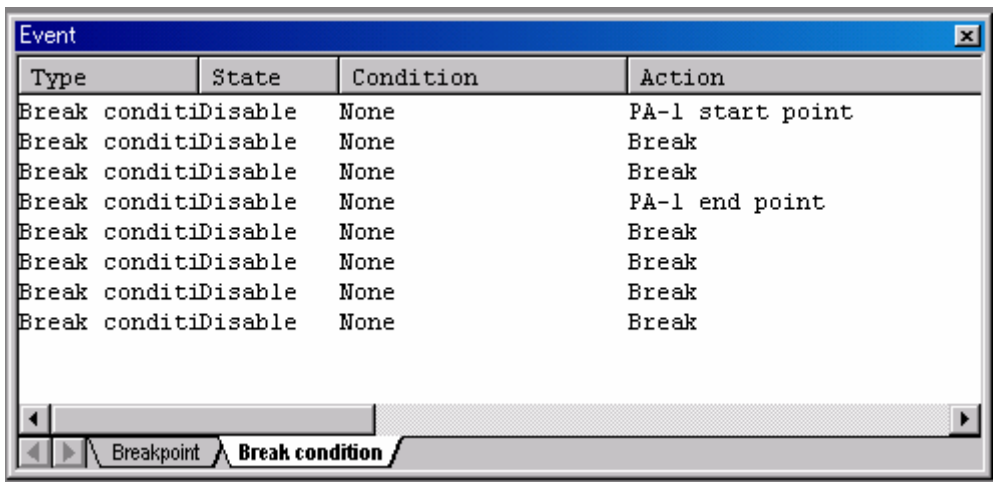


Figure 2.4 [Event] Window ([Break condition] Sheet)

In this case, break will not occur when the conditions of Break Condition 1 and Break Condition 4 are satisfied.

Note: When the range is specified, be sure to set the measurement start and end conditions for Break Condition 1 and Break Condition 4, respectively, and then execute the user program. If Break Condition 1 or Break Condition 4 is not set and the user program is executed, performance is not measured normally. In this case, the following dialog box is displayed.

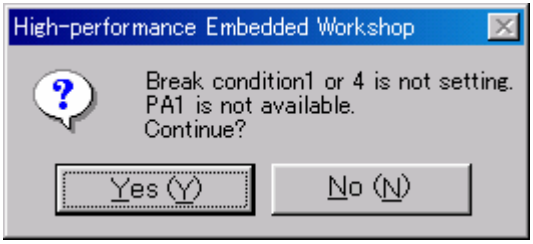


Figure 2.5 [High-performance Embedded Workshop] Dialog Box

(b) Measurement item

Items are measured with [Channel 1 to 2] in the [CPU Performance] dialog box. Maximum two conditions can be specified at the same time. Table 2.7 shows the measurement items (Options in table 2.7 are parameters for <mode> of the PERFORMANCE_SET command. They are displayed for NAME in the [Performance Analysis] window).

Table 2.7 Measurement Items

Event	Keyword	Description
Operand access count (read/with cache)	OAR*	The number of times the operand access is performed on the cacheable area when the cache is enabled (read access only).
Operand access count (write/with cache)	OAW*	The number of times the operand access is performed on the cacheable area when the cache is enabled (write access only).
Operand access count (read and write/with cache)	OARW*	The number of times the operand access is performed on the cacheable area when the cache is enabled (both read and write accesses).
Internal RAM operand access count	OARAM	The number of times the internal RAM area is accessed.
All operand access count	OA	The number of all operand accesses.
Internal I/O area access count	IOA	The number of times the internal I/O area is accessed.
Operand cache read miss count	DCR	The number of times operand cache misses occur at data reading.
Operand cache write miss count	DCW	The number of times operand cache misses occur at data writing.
Operand cache read and write miss count	DCRW	The number of times operand cache misses occur at data reading or writing.
Instruction cache miss count	EC	The number of times instruction cache misses.
UTLB miss count	DT	The number of times UTLB misses occur at data access.
Instruction TLB miss count (ITLB and UTLB misses)	ET	The number of times UTLB and ITLB misses occur at instruction access.
Instruction cache miss count	EF*	The number of times instructions are fetched from the cacheable area when the cache is enabled.
All instruction fetch count	EA	The number of times all instructions are fetched.
Branch instruction execution count	B	The number of times branch instructions are issued (instructions to be counted: BF (other than displacement 0), BF/S and BT (other than displacement 0), BT/S, BRA, BRAF, and JMP).
Branch taken count	BT	The number of times branches are taken (branches to be counted are the same as mode B).
BSR/BSRF/JSR instruction execution count	BBJ	The number of times the BSR, BSRF, or JSR instruction is issued.
Instruction execution count	E	The number of times instructions are issued.

Table 2.7 Measurement Items (cont)

Event	Keyword	Description
Two-instruction concurrent execution count	E2	The number of times two instructions are issued at the same time.
FPU instruction execution count	EFP	The number of times FPU instruction is issued.
TRAPA instruction execution count	ETR	The number of times the TRAPA instruction is executed.
Interrupt count (normal)	INT	The number of interrupts (generally except for NMI).
Interrupt count (NMI)	NMI	The number of NMI interrupts.
UBC-A match count	UA	The number of times channel A of UBC is satisfied.
UBC-B match count	UB	The number of times channel B of UBC is satisfied.
Instruction cache-fill cycle	ECF	The number of instruction cache-fill cycles.
Operand cache-fill cycle	OCF	The number of operand cache-fill cycles.
Elapsed-time cycle	TM	The number of cycles for elapsed time.
Pipeline-freeze (by cache miss or instruction)	PFCE	Pipeline-freeze cycle due to instruction cache misses.
Pipeline-freeze (by cache miss or data)	PFCD	Pipeline-freeze cycle due to operand cache misses.
Pipeline-freeze (by branch instruction/interrupt)	PFB	Pipeline-freeze cycle due to branch instructions or exceptions.

Note: For the non-cache operand accesses due to the PREF instruction or TLB.c=0, the correct value cannot be counted.

The events can be counted even in the conditions shown in table 2.8, in addition to the normal count conditions.

Table 2.8 Performance Count Conditions

Event	Count Condition	Target Mode
Instruction cache miss count	• Includes instruction fetch for the cache-off area to count the number of times the instruction has not been fetched in one cycle. • When a cache miss occurs during an overrun fetch generated at exception.	EC
TLB miss count	When the TLB miss is canceled by an exception having a higher priority than that of the TLB miss	DT and ET
Instruction fetch count	When the instruction fetch request by the CPU is accepted.	EF and EA
Instruction issue count	Counts one when two instructions are issued at the same time.	E
	Counts one to three when instruction fetch exception (instruction address error, instruction TLB miss exception, or instruction TLB protection violation exception) occurs.	E and E2
FPU instruction issue count	• Counts one when two instructions are issued at the same time. • The following shows the FPU instructions: LDS Rm, FPUL, LDS.L @Rm+, FPUL, LDS Rm, FPSCR, LDS.L @Rm+, FPSCR, STS FPUL, Rn, STS.L FPUL, @-Rn, STS FPSCR, Rn, STS.L FPSCR, @-Rn Others: instructions that the instruction code is H'Fxxx	EFP
UBC satisfaction count	Also counts when the emulator uses the UBC as Break Condition 5,6.	UA and UB
Pipeline freeze due to cache miss	Includes the following freeze times: • At internal RAM or internal I/O space access • At instruction or operand access without cache	PFCE and PFCD
Pipeline freeze cycle due to branch instruction or exception	Counts only one cycle at branch instruction execution except when the delay slot instruction is executed with one-cycle delay. One instruction is executed in one cycle, which is similar to the branch count. When the instruction in the branch destination does not exist in the instruction cache, the delay after the second cycle is counted by the ECF. In the PFB, all branch instructions can be counted.	PFB

(c) Counting method

One of the following methods can be specified by each of measurement channels 1 and 2.

1. Counted by the CPU operating clock
2. Counted by the ratio of the CPU operating clock to the bus clock

When the above method 1 is specified, one CPU operating clock cycle is counted as one. When method 2 is specified, the count is incremented by 3, 4, 6, 8, 12, or 24, according to the clock frequency ratio (ratio of the CPU clock to the bus clock). In this case, the execution time can be calculated by the following expression:

T = C x B / 24 (T: Execution time; B: Time of one bus clock cycle; C: Count)

When the ratio of the CPU clock to the bus clock is changed in the user program, it is recommended to select method 2, above, to count the number of cycles.

The following shows examples to measure the performance of the user program by the performance measurement function.

(i) Measuring cache hit ratio

Specify measurement channel 1 to count the cache misses (for data read and write) and specify measurement channel 2 to count operand accesses (read and write) to the cacheable area while the cache is enabled. Specify, with both the channels, the measurement from the start to the end of user program execution.

With the above command settings, the cache miss count and the access count to the cacheable area can be measured, and the cache hit ratio in the executed user program can be obtained.

(ii) Measuring ratio of execution time in specified program area to total execution time

Specify measurement channel 1 to measure the elapsed cycle count from the start to the end of user program execution. Specify measurement channel 2 to measure the elapsed cycle count during execution from the specified start PC to the specified end PC.

With both the channels, the total elapsed cycle and the elapsed cycle for the specified program area can be measured, and the ratio of the execution time in the specified program area to the total execution time can be obtained.

Notes: 1. The counter for performance measurement has 48 bits. A maximum of $2^{48} = 2.8 \times 10^{14}$ counts and 21.7-day cycles (when the CPU operating frequency is 267 MHz) can be measured. If a counter overflow occurs, the count becomes invalid.

2. For details on command-line syntax, refer to the online help.

2. Displaying the measured result

   The measured result is displayed in the [Performance Analysis] window or the PERFORMANCE_ANALYSIS command with hexadecimal (32 bits).

Note: If a performance counter overflows as a result of measurement, "********" will be displayed.

3. Initializing the measured result

   To initialize the measured result, select [Initialize] from the popup menu in the [Performance Analysis] window or specify INIT with the PERFORMANCE_ANALYSIS command.

2.2.7 Interrupts

During emulation, any interrupt to the SH7750R can be used. Whether or not to process interrupts during emulator command execution or in command input wait state can be specified.

— When no interrupt is processed during user program break

  While the emulator is executing the user program or is in command input wait state, interrupts are not processed generally. However, if an internal interrupt or an edge sensitive external interrupt occurs in command input wait state, the emulator holds the interrupt and executes the interrupt processing routine when the GO command is entered.

— When interrupts are processed during user program break

  Use the INTERRUPT command to execute an interrupt during a user program break. This function is supported only with the command lines.

  - Execute only non-maskable interrupt (NMI)
  - Sets the priority and executes only interrupts with high priority

Notes: 1. Check that the interrupt handler operates normally before using this function. In addition, do not execute a non-limited loop or the sleep instruction in the interrupt handler. If the processing of the handler does not end, the emulator generates a Communication Timeout error.

2. When interrupts are accepted during user program break, user interrupt processing is not traced. In this case, continuous trace is not enabled.
3. Use the NOP instruction at the delay slot after the RTE instruction in the interrupt handler.
4. If a user interrupt is inserted while the user program breaks until the processing ends, do not set a BREAKPOINT in the interrupt handler. The emulator may generate a Communication Timeout error. Use the Break Condition function.
5. For details on command-line syntax, refer to the online help.

2.2.8 CPU Status Acquisition

The emulator can display the SH7750R status during user program execution in realtime. It displays the items selected in the [Extended Monitor Configuration] dialog box in the [Extended Monitor] window during user program execution. The emulator can display the state of the moment when a command is input for the specified register through the command-line function.

Notes: 1. This function is only valid during user program execution. If this function is used during a user program break, an undefined value is displayed.
2. A read value during reset is not guaranteed.
3. In the sleep or deep sleep mode, only the STATUS or FRQCR can be read.
4. The display is updated in the 1000- to 65535-ms interval.

Table 2.9 shows the details of the items that can be displayed.

Table 2.9 Display Status

Item	Example	Description
PC	H'A0000104	Displays the PC value.
SR	H'000000F0	Displays the SR register value.
FPSCR	H'000000F0	Displays the FPSCR register value.
INTEVT	H'00000100	Displays the INTEVT register value.
EXPEVT	H'00000600	Displays the EXPEVT register value.
FRQCR register	H'00000102	Displays the FRQCR register value.
MMUCR.AT	H'0	Displays the AT bit value in the MMUCR register.
ASID	H'01	Displays the ASID value in the PTEH register.
CCR	H'00000001	Displays the CCR register value.
SBUS	H'00000000	Displays the load/store bus address. (internal bus)
EBUS	H'0000000	Displays the external bus address.
SBTYPE	B'1101	Displays the internal bus state. Each bit has the following meanings: **Bit3: Bus access** 0: Without bus access 1: With bus access If bit 3 is 0, other bits of SBTYPE and all bits of SBUS are invalid.

Table 2.9 Display Status (cont)

Item	Example	Description
SBTYPE (cont)	B'1101	**Bit2: Read or write cycle** 0: Read cycle 1: Write cycle **Bit1,0: Bus width** Bit1=0, Bit0=0: 8-bit bus width Bit1=0, Bit0=1: 16-bit bus width Bit1=1, Bit0=0: 32-bit bus width Bit1=1, Bit0=1: 64-bit bus width
EBTYPE	B'0000000	Displays the external bus state. Each bit has the following meanings: **Bit5: Bus mode at DMA transfer** Displays an invalid value in the CPU access. 0: Burst mode 1: Cycle steal mode **Bit4: CPU access or DMAC access** 0: Access from CPU 1: Access from DMAC **Bit6,3,2: One transfer unit in DMA transfer** Bit6=0, Bit3=0, Bit2=0: 64 bits Bit6=1, Bit3=0, Bit2=0: 32 bytes Bit6=0/1, Bit3=0, Bit2=1: 8 bits Bit6=0/1, Bit3=1, Bit2=0: 16 bits Bit6=0/1, Bit3=1, Bit2=1: 32 bits These bits indicate memory access in the chip instead of the bus width. **Bit1: Read or write cycle** 0: Read cycle 1: Write cycle **Bit0: Bus access** 0: Without bus access 1: With bus access If bit 0 is 0, other bits of EBTYPE and all bits of EBUS are invalid. Note: When bit 0 is 1 and bit 4 is 0, bits 5 and 6 become invalid.
STATUS	B'00	Displays the STATUS pin state.

Table 2.9 Display Status (cont)

Item	Example	Description
Condition match flag	A=0	Displays whether the channel A condition of the UBC has been satisfied. When the UBC is used as a Break Condition, it displays whether Break Condition 7 has been satisfied. 0: Not satisfied 1: Satisfied
	B=0	Displays whether the channel B condition of the UBC has been satisfied. When the UBC is used as a Break Condition, it displays whether Break Condition 6 has been satisfied. 0: Not satisfied 1: Satisfied
	BC1=0	Displays whether Break Condition 1 has been satisfied. 0: Not satisfied 1: Satisfied
	BC2=0	Displays whether Break Condition 2 has been satisfied. 0: Not satisfied 1: Satisfied
	BC3=0	Displays whether Break Condition 3 has been satisfied. 0: Not satisfied 1: Satisfied
	BC4=0	Displays whether Break Condition 4 has been satisfied. 0: Not satisfied 1: Satisfied
Condition match flag for sequential break	A=0	When the sequential break condition of the UBC is selected, this bit is 1 when the channel A condition has been satisfied and the channel B condition has not been satisfied. When the UBC is used as a Break Condition, channel A and channel B correspond to Break Condition 7 and Break Condition 6, respectively. This bit is 1 when Break Condition 7 has been satisfied and Break Condition 6 has not been satisfied.
	BC4=0	When Sequential break condition 4-3-2-1 is selected, this bit is 1 when Break Condition 4 has been satisfied and Break Condition 3 has not been satisfied. It is also 1 when Break Condition 4 is satisfied again after Break Condition 3 has been satisfied.

Table 2.9 Display Status (cont)

Item	Example	Description
Condition match flag for sequential break (cont)	BC3=0	When Sequential break condition 4-3-2-1 and Sequential break condition 3-2-1 are selected, this bit is 1 when Break Condition 3 has been satisfied and Break Condition 2 has not been satisfied. It is also 1 when Break Condition 3 is satisfied again after Break Condition 2 has been satisfied.
	BC2=0	When Sequential break condition 4-3-2-1, Sequential break condition 3-2-1, and Sequential break condition 2-1 are selected, this bit is 1 when Break Condition 2 has been satisfied and Break Condition 1 has not been satisfied. It is also 1 when Break Condition 2 is satisfied again after Break Condition 1 has been satisfied.

(a) Window function

To use the window function, the [Extended Monitor] window is displayed.

Open the [Extended Monitor] window by selecting [Display -> CPU -> Extended Monitor] or click the [Extended Monitor] toolbar button ( ).

Set the items to be displayed by selecting [Property] from the popup menu with the right mouse button to display the [Extended Monitor Configuration] dialog box.

Select the check boxes in the [Settings] group box for the items that are to be displayed.

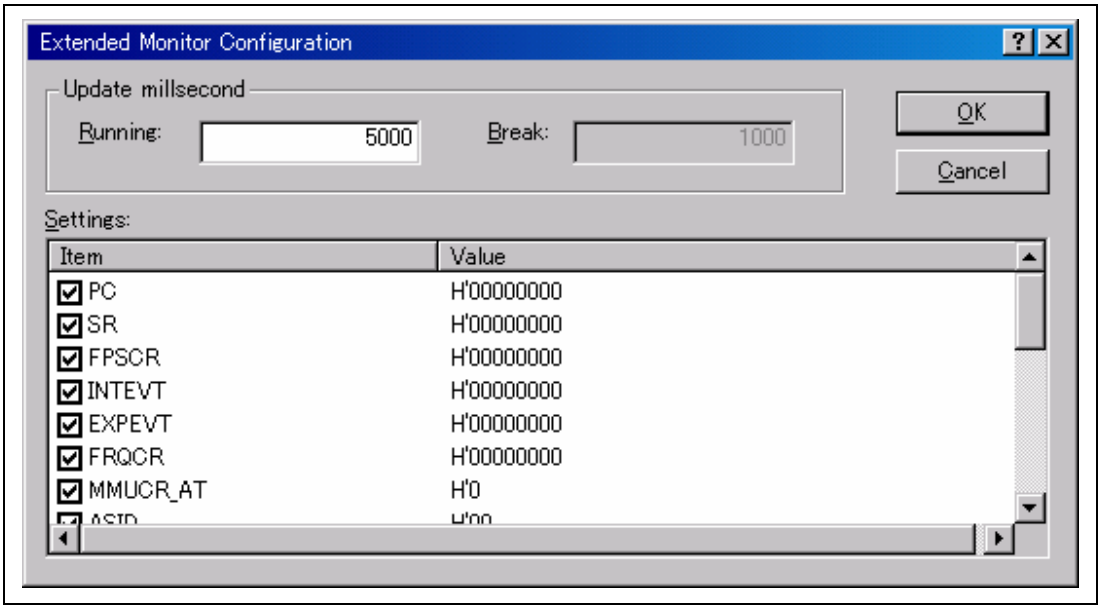


Figure 2.6 [Extended Monitor Configuration] Dialog Box

The items that have been selected are displayed in the [Extended Monitor] window.

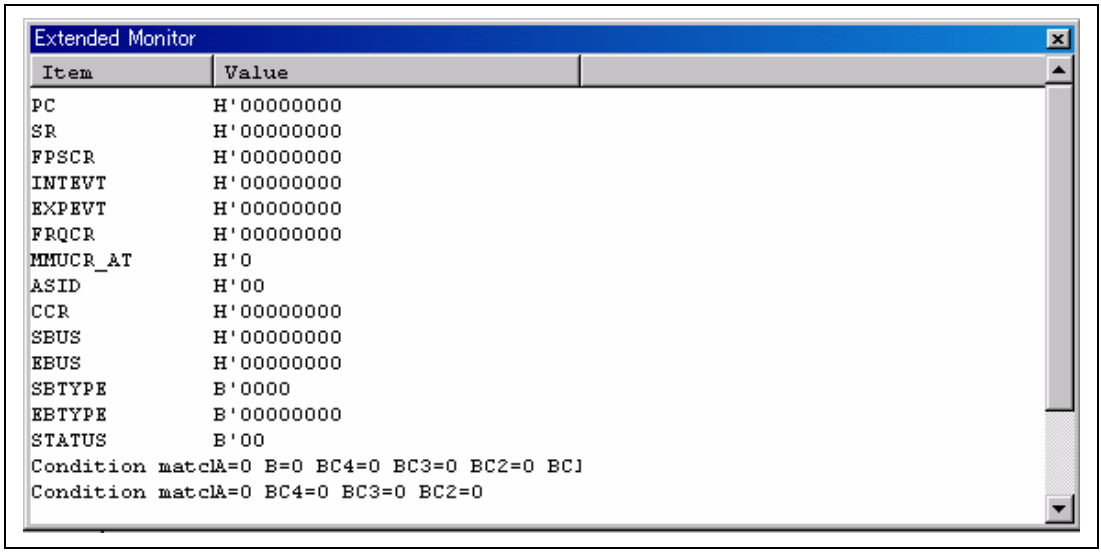


Figure 2.7 [Extended Monitor] Window

Notes: 1. CPU status acquisition function [Condition match flag]:
The Break Condition function clears the condition match flag after a break occurred. Therefore, note that there are following limitations on measurement of this function.

Break Condition 1,4: Have meaning when they are used as the measurement start/end condition in the performance measurement function. In other cases, they are invalid in this product.

Break Condition 2,3,5,6: Are invalid in this product.

When Break Condition 5,6 are used as the UBC: The condition match flag is 1 when each channel in the UBC is satisfied until the flag is cleared.

2. CPU status acquisition function during standby:
The read value during standby cannot be guaranteed.

SuperH™ Family E10A-USB Emulator
Additional Document for User's Manual
Supplementary Information on Using the SH7750R

Publication Date: Rev.1.00, March 15, 2004
Rev.2.00, March 19, 2007
Published by: Sales Strategic Planning Div.
Renesas Technology Corp.
Edited by: Customer Support Department
Global Strategic Communication Div.
Renesas Solutions Corp.

©2007. Renesas Technology Corp., All rights reserved. Printed in Japan.

Renesas Technology Corp. Sales Strategic Planning Div. Nippon Bldg., 2-6-2, Ohte-machi, Chiyoda-ku, Tokyo 100-0004, Japan

RENESAS SALES OFFICES

http://www.renesas.com

Refer to "**http://www.renesas.com/en/network**" for the latest and detailed information.

Renesas Technology America, Inc.
450 Holger Way, San Jose, CA 95134-1368, U.S.A
Tel: <1> (408) 382-7500, Fax: <1> (408) 382-7501

Renesas Technology Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K.
Tel: <44> (1628) 585-100, Fax: <44> (1628) 585-900

Renesas Technology (Shanghai) Co., Ltd.
Unit 204, 205, AZIACenter, No.1233 Lujiazui Ring Rd, Pudong District, Shanghai, China 200120
Tel: <86> (21) 5877-1818, Fax: <86> (21) 6887-7898

Renesas Technology Hong Kong Ltd.
7th Floor, North Tower, World Finance Centre, Harbour City, 1 Canton Road, Tsimshatsui, Kowloon, Hong Kong
Tel: <852> 2265-6688, Fax: <852> 2730-6071

Renesas Technology Taiwan Co., Ltd.
10th Floor, No.99, Fushing North Road, Taipei, Taiwan
Tel: <886> (2) 2715-2888, Fax: <886> (2) 2713-2999

Renesas Technology Singapore Pte. Ltd.
1 Harbour Front Avenue, #06-10, Keppel Bay Tower, Singapore 098632
Tel: <65> 6213-0200, Fax: <65> 6278-8001

Renesas Technology Korea Co., Ltd.
Kukje Center Bldg. 18th Fl., 191, 2-ka, Hangang-ro, Yongsan-ku, Seoul 140-702, Korea
Tel: <82> (2) 796-3115, Fax: <82> (2) 796-2145

Renesas Technology Malaysia Sdn. Bhd
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No.18, Jalan Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: <603> 7955-9390, Fax: <603> 7955-9510

SuperH™ Family E10A-USB Emulator Additional Document for User's Manual Supplementary Information on Using the SH7750R

Renesas Technology Corp.
2-6-2, Ote-machi, Chiyoda-ku, Tokyo, 100-0004, Japan